Chapter 20

Agent Extrinsic Safety: Interaction Risks

As AI agents evolve and interact with increasingly complex environments, the safety risks associated with these interactions have become a critical concern. This chapter focuses on AI agent's engagement with memory systems, physical and digital environments, and other agents. These interactions expose AI agents to various vulnerabilities, ranging from memory corruption and environmental manipulation to adversarial behavior in multi-agent systems. By examining these interaction risks, we aim to highlight the diverse threats that can undermine the integrity and reliability of AI agents in real-world applications. The following sections explore these challenges in detail, discussing specific attack vectors and their implications for system safety.

20.1 Agent-Memory Interaction Threats

The extrinsic memory module functions as the cognitive repository that empowers intelligent agents to store, retrieve, and contextualize information, facilitating continuous learning and the execution of complex tasks through accumulated experiences. Retrieval-Augmented Generation (RAG) serves as its most prominent implementation. However, RAG frameworks are vulnerable to adversarial manipulations that deceive agents into retrieving and utilizing harmful or misleading documents. AgentPoison [1194] exploits this vulnerability by executing a backdoor attack on AI agents, poisoning RAG knowledge bases to ensure that backdoor-triggered inputs retrieve malicious demonstrations while maintaining normal performance on benign queries. ConfusedPilot [1353] exposes a class of RAG system vulnerabilities that compromise the integrity and confidentiality of Copilot through prompt injection attacks, retrieval caching exploits, and misinformation propagation. Specifically, these attacks manipulate the text input fed to the LLM, causing it to generate outputs that align with adversarial objectives. PoisonedRAG [1354] represents the first knowledge corruption attack on RAG, injecting minimal adversarial texts to manipulate LLM outputs. Framed as an optimization problem, it achieves a 90% success rate with just five poisoned texts per target question in large databases. Jamming [1355] introduces a denial-of-service attack on RAG systems, where a single adversarial "blocker" document inserted into an untrusted database disrupts retrieval or triggers safety refusals, preventing the system from answering specific queries. BadRAG [1356] exposes vulnerabilities in RAG-based LLMs through corpus poisoning, wherein an attacker injects multiple crafted documents into the database, forcing the system to retrieve adversarial content and generate incorrect responses to targeted queries. By introducing just 10 adversarial passages (0.04% of the corpus), it achieves a 98.2% retrieval success rate, elevating GPT-4's rejection rate from 0.01% to 74.6% and its negative response rate from 0.22% to 72%. TrojanRAG [1357] executes a joint backdoor attack on RAG systems, optimizing multiple backdoor shortcuts via contrastive learning and enhancing retrieval with a knowledge graph for fine-grained matching. By systematically normalizing backdoor scenarios, it evaluates real-world risks and the potential for model jailbreak. Lastly, a covert backdoor attack [1358] leverages grammar errors as triggers, allowing LLMs to function normally for standard queries while retrieving attacker-controlled content when minor linguistic mistakes are present. This method exploits the sensitivity of dense retrievers to grammatical irregularities using contrastive loss and hard negative sampling, ensuring that backdoor triggers remain imperceptible while enabling precise adversarial control.

20.2 Agent-Environment Interaction Threats

Agents can be classified into two categories based on their mode of interaction: physical interaction agents and digital interaction agents. Physical interaction agents operate in the real world, using sensors and actuators to perceive and

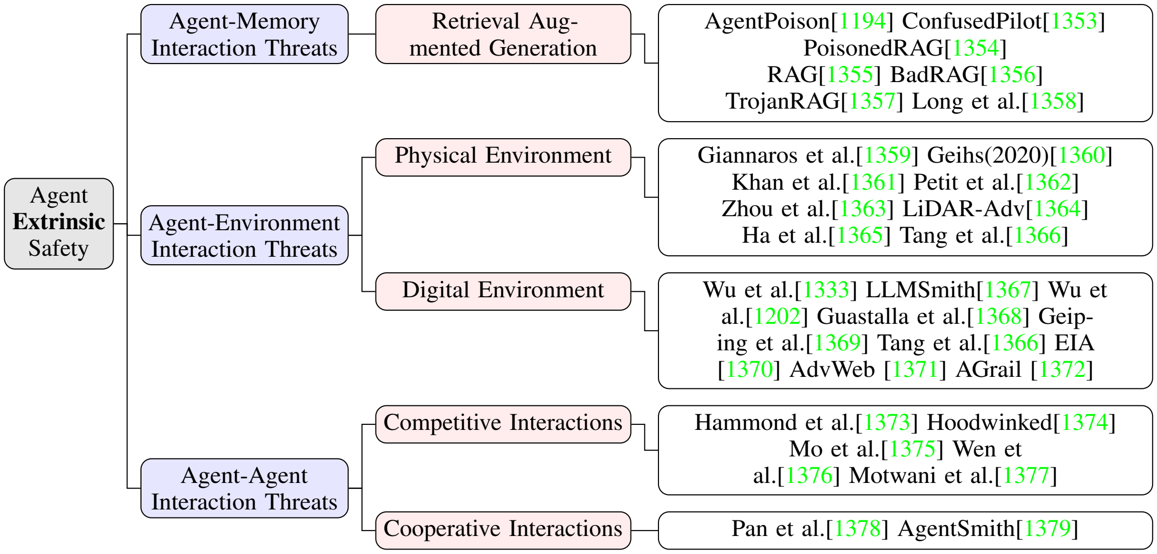
Agent Extrinsic Safety
Agent-Memory Interaction Threats
Retrieval Augmented Generation
AgentPoison[1194] ConfusedPilot[1353] PoisonedRAG[1354] RAG[1355] BadRAG[1356] TrojanRAG[1357] Long et al.[1358]
Agent-Environment Interaction Threats
Physical Environment
Giannaros et al.[1359] Geihs(2020)[1360] Khan et al.[1361] Petit et al.[1362] Zhou et al.[1363] LiDAR-Adv[1364] Ha et al.[1365] Tang et al.[1366]
Digital Environment
Wu et al.[1333] LLMSmith[1367] Wu et al.[1202] Guastalla et al.[1368] Geiping et al.[1369] Tang et al.[1366] EIA [1370] AdvWeb [1371] AGrail [1372]
Agent-Agent Interaction Threats
Competitive Interactions
Hammond et al.[1373] Hoodwinked[1374] Mo et al.[1375] Wen et al.[1376] Motwani et al.[1377]
Cooperative Interactions
Pan et al.[1378] AgentSmith[1379]

Figure 20.1: Agent Extrinsic Safety: Threats on agent-memory, agent-environment, and agent-agent interactions.

influence their environment. Examples of such agents include autonomous vehicles and robotic systems. In contrast, digital interaction agents function within virtual or networked environments, processing and responding to data from digital sources. These include AI-powered chatbots, cybersafety systems, and automated trading algorithms.

Threats in Physical Environment. Agents operating in the physical world, such as robots and autonomous vehicles, face distinct safety challenges due to their interaction with dynamic and potentially adversarial environments [1359, 1360, 1366]. One major threat is sensor spoofing, where attackers manipulate sensor inputs to deceive the agent about its surroundings. For example, GPS spoofing can pose significant risks to UAVs (unmanned aerial vehicles) and other GPS-dependent platforms by misleading autonomous vehicles about their actual location. This allows for malicious redirection or hijacking [1361]. Similarly, LiDAR spoofing can introduce false obstacles that don't actually exist, potentially leading to navigation failures or safety hazards [1362]. Another critical risk is actuator manipulation, where adversaries take control of an agent's actuators, forcing it to perform unintended physical actions. This can occur through direct tampering with the hardware or by exploiting vulnerabilities in the software that governs actuator functions [1363]. Such attacks can compromise the agent's actions, leading to physical harm or mission failure. Additionally, exploiting environmental hazards is a serious threat. Attackers may introduce physical obstacles or manipulate environmental conditions to disrupt an agent's operations. For example, adversarial objects created using techniques like LiDAR-Adv can deceive LiDAR-based autonomous driving systems by inducing sensor misinterpretations, thus degrading detection reliability and increasing real-world safety risks [1364]. Lastly, misalignment in physical actions can undermine the safety of autonomous agents. Discrepancies between an agent's perception and the actual physical constraints of its environment can lead to unsafe or infeasible actions. For example, mismatches between learned locomotion policies and real-world physics—such as misjudging terrain rigidity or obstacle dimensions—can cause autonomous agents to take hazardous steps (e.g., unstable strides on rough surfaces). This has been observed in prior systems that required over 100 manual resets due to uncontrolled falls [1365].

Threats in Digital Environment. Agents operating in digital environments, such as software agents and web-based agents, face distinct safety challenges arising from their reliance on external data sources and computational resources [1333, 1366]. One major threat is code injection, where malicious actors introduce harmful code into the agent's environment, leading to unintended command execution [1367]. These attacks often exploit software vulnerabilities or leverage compromised external resources that the agent interacts with, potentially resulting in unauthorized control over the agent's operations [1202]. Environmental Injection Attack (EIA) exploits privacy risks in generalist web agents to stealthily steal users' PII, achieving up to 70% success rate [1370]. AdvWeb is an automated adversarial prompt generation framework to mislead black-box web agents into executing harmful actions [1371]. Another critical risk is data manipulation, where attackers alter the information an agent receives, causing incorrect decisions or actions [1333]. For example, a trading agent can be misled by manipulated financial data, leading to incorrect transactions, or an information-gathering agent may be tricked by falsified news articles, distorting its outputs. Such manipulations can have cascading effects, especially in automated systems that rely on accurate data for decision-making. Beyond direct manipulation, denial-of-service (DoS) attacks pose a serious threat by overwhelming the agent's digital environment with excessive requests or data, effectively rendering it unresponsive or causing it to crash [1368]. These disruptions can be particularly detrimental to time-sensitive applications where availability and responsiveness are critical. Additionally, resource exhaustion is a significant threat, as adversaries may exploit the agent's resource management mechanisms to deplete computational resources, leading to service denial for other users or overall system instability [1369]. By draining processing power, memory, or bandwidth, attackers can severely impair an agent's ability to function effectively, disrupting its operations and reducing its efficiency. In addressing the safety challenges of LLM agents, AGrail is proposed as a lifelong guardrail framework that enhances agent security by adapting safety checks to mitigate task-specific and systemic risks, demonstrating robust performance and transferability across diverse tasks [1372].

20.3 Agent-Agent Interaction Threats

In multi-agent systems, interactions between agents can introduce new safety vulnerabilities [1380]. These interactions are mainly competitive, where agents try to outdo each other, or cooperative, where they work together.

Threats in Competitive Interactions. When agents compete, they often use tricky methods to gain an advantage [1373]. For example, they might spread false information or make other agents think the situation is different from reality to deceive them [1374]. This can lead opponents to make poor decisions, weakening their position. Apart from misinformation, agents may also try to take advantage of weaknesses in their opponent's algorithms or strategies [1375]. By identifying these weaknesses, they can predict and manipulate the other agent's behavior, gaining an edge in the competition. Additionally, some agents might use disruptive techniques like denial-of-service (DoS) attacks, which overload an opponent's system with unnecessary requests, disrupting communication and hindering their ability to function [1376]. Another threat in competitive interactions is covert collaboration. Sometimes agents secretly cooperate, even when it's against the rules, to manipulate the outcome in their favor [1377]. This kind of collusion undermines fairness and damages the integrity of the system, as it skews the competition in their favor.

Threats in Cooperative Interactions. In cooperative situations, where agents work together toward a common goal, safety threats could damage the system's stability and reliability. One risk is unintentional information leakage, where agents accidentally share sensitive data during their communication. This could lead to privacy violations or unauthorized access, weakening the system's trustworthiness. In addition to data leaks, errors made by one agent can spread throughout the system, causing bigger failures and lowering overall performance. [1378] discusses this problem in Open-Domain Question Answering Systems (ODQA), where errors from one part of the system can ripple through and affect other components, severely impacting reliability. The situation becomes even worse if one compromised agent introduces a vulnerability that spreads to others. If a hacker successfully takes control of one agent, they could exploit weaknesses throughout the entire system, leading to a major safety failure [1379]. This kind of widespread compromise is dangerous because it could start with a small breach and escalate quickly. Another challenge comes from poor synchronization between agents. If agents don't update their information at the same time or experience delays in communication, it can cause problems in decision-making. Misalignment or delays in updates can disrupt coordination, making it harder for the agents to achieve their shared goals effectively. These challenges emphasize the need for strong safety systems in cooperative multi-agent setups to keep them reliable and resistant to attacks.

20.4 Summary and Discussion

The preceding sections have detailed the significant safety risks that arise from AI agents interacting with memory systems, physical and digital environments, and other agents. These risks, ranging from data poisoning and code injection to sensor spoofing and collusion, highlight the vulnerabilities inherent in increasingly complex agent-based systems. However, as AI agents become more capable, utilizing natural language understanding and specialized tools for sophisticated reasoning, researchers are actively developing safety protocols to address these challenges. These protocols differ in approach for general-purpose and domain-specific agents.

General-purpose agents, designed for versatility across various domains, face a broad spectrum of safety challenges. To mitigate these risks, researchers have developed several methods to enhance their safety. Evaluation mechanisms, such as AgentMonitor [1381], assess the safety awareness of agents by monitoring their decision-making processes and identifying potentially unsafe actions. R-Judge [1382] quantifies an agent's risk awareness by evaluating its responses to both malicious and benign queries, offering a systematic approach to safety compliance. Additionally, risk detection tools like ToolEmu [795] simulate tool usage in controlled environments to expose vulnerabilities in agent interactions. This approach identifies potential hazards during task execution, allowing developers to address vulnerabilities proactively. These combined efforts enhance the safety of general-purpose agents through comprehensive evaluation and risk detection.

Domain-specific agents, tailored for specialized tasks in high-stakes environments like scientific research, require even more stringent safety measures. Safety tools such as ChemCrow [1383] are designed to mitigate risks in chemical synthesis tasks by reviewing user queries and filtering malicious commands, ensuring agents do not inadvertently synthesize hazardous chemicals. Structured task constraints, as implemented in CLAIRify [1384], enhance experimental safety by imposing high-level constraints on material synthesis order and low-level restrictions on manipulation and perception tasks, thereby preventing accidents and errors. Furthermore, benchmarks like SciGuard [1385], which includes the SciMT-Safety benchmark, evaluate model safety by measuring both harmlessness (rejecting malicious queries) and helpfulness (handling benign queries effectively). SciGuard also incorporates long-term memory to enhance agents' ability to safely execute complex instructions while maintaining accurate risk control. These focused approaches ensure that domain-specific agents operate safely and effectively within their specialized fields.

In summary, significant progress has been made in developing innovative evaluation mechanisms and risk mitigation strategies to enhance the safety of both general-purpose and domain-specific AI agents. However, a critical area for future research lies in integrating these approaches. Building stronger connections between the broad capabilities of general-purpose agents and the focused safeguards of domain-specific agents will be essential for creating truly robust and trustworthy LLM systems. The challenge is to combine the best aspects of both approaches to develop agents that are both versatile and secure.